CODED CACHING AND WIRELESS COMMUNICATIONS

PETROS ELIA

(EURECOM – FRANCE)

Intro

- This talk aims to discuss two aspects:
  - A novel caching method (Coded caching: yields very substantial gains)
  - In wireless communication networks (Why is wireless really different?)

- New paradigm: Using caches
  - NOT to reduce the volume/size of the problem
    - *"Prefetch something today so that you don't have to send it tomorrow"*
  - BUT to surgically alter the informational structure of networks
    - Use caches to change the network to something faster and simpler

Outline

- Basic elements of coded caching
  - Basic properties
  - Main gains
- Important variants
  - File popularity statistics
  - Schemes with reduced subpacketization

Outline

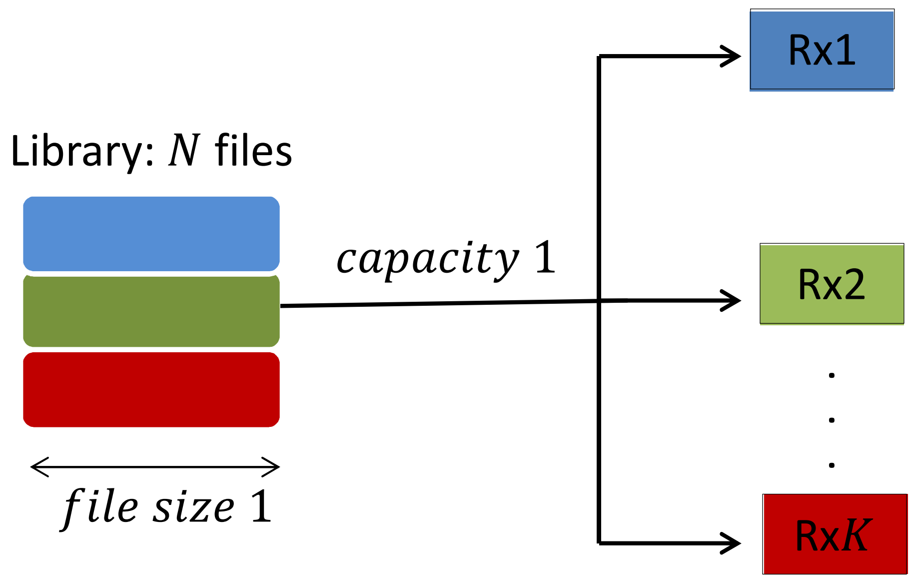
- Need to fuse coded caching with advanced PHY techniques

- Exploring/exploiting salient features of wireless w.r.t. caching
  - XORs in the air
  - MIMO
  - Feedback
  - Non linearities
  - Topology
  - Channel fluctuations
  - Spatial reuse…

- Theoretical and practical open problems/bottlenecks

Simple Caching

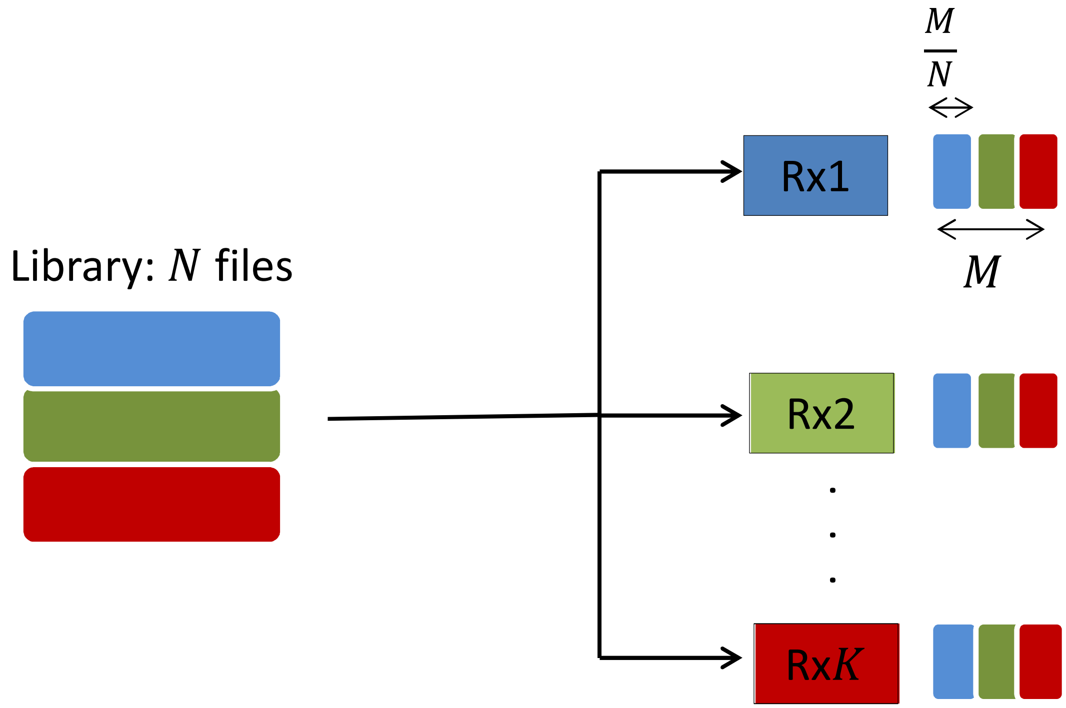
Single stream channel: No caching



- Transmission sequence:

$$T = K$$

Simple caching (uniform popularity – for now)

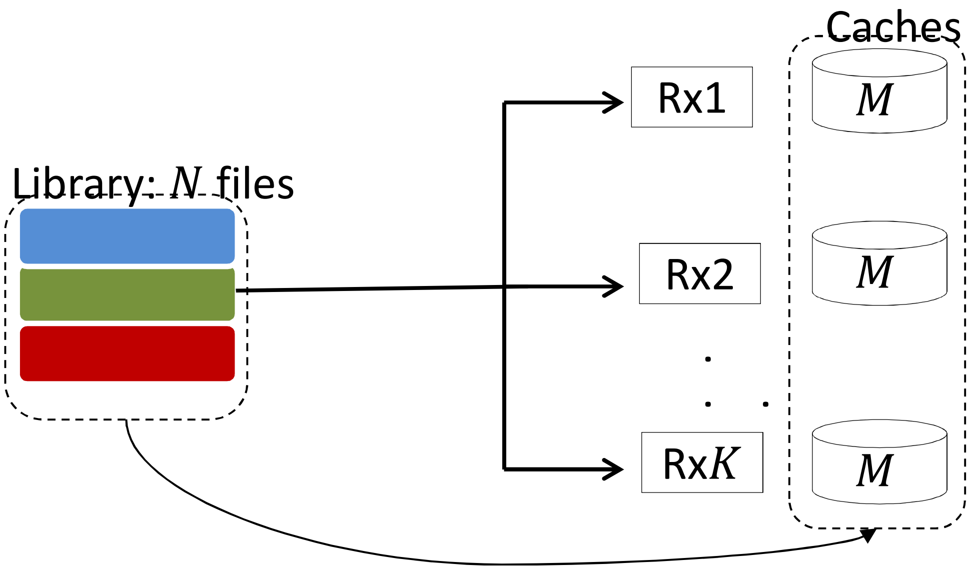


- Transmission sequence: 

- Local cache gain: $(1 - M/N)$ for each user
- The rate:

$$T = K(1 - M/N) = K(1 - \gamma), \qquad \gamma \stackrel{\text{def}}{=} \frac{M}{N}$$

Basic Parameters

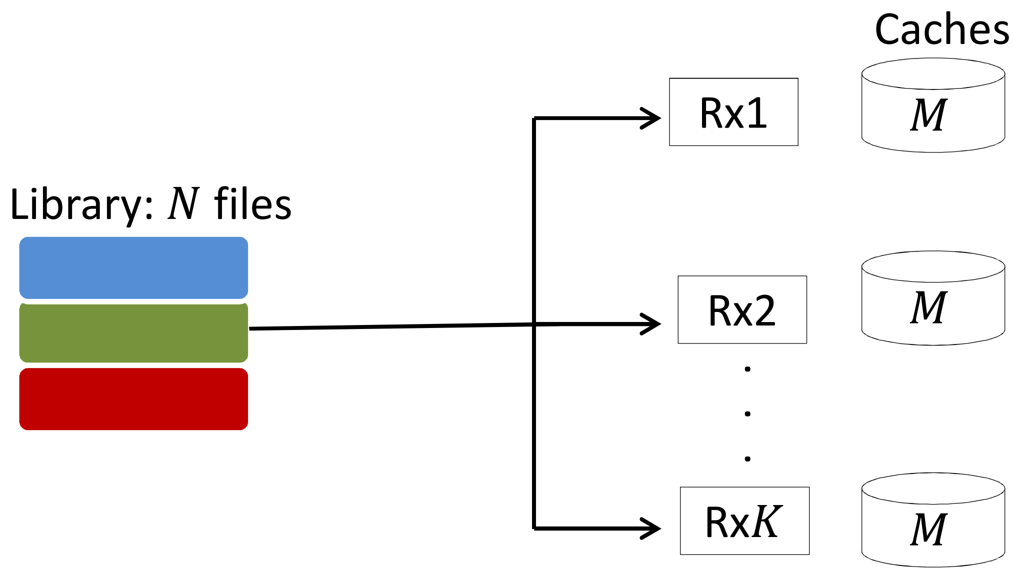


$$\gamma \overset{\text{def}}{=} \frac{M}{N} \overset{\text{def}}{=} \frac{indivial\ cache\ size}{library\ size}$$

$T(\gamma)$: *duration of delivery phase*

OBJECTIVE: reduce $T(\gamma)$

Coded caching

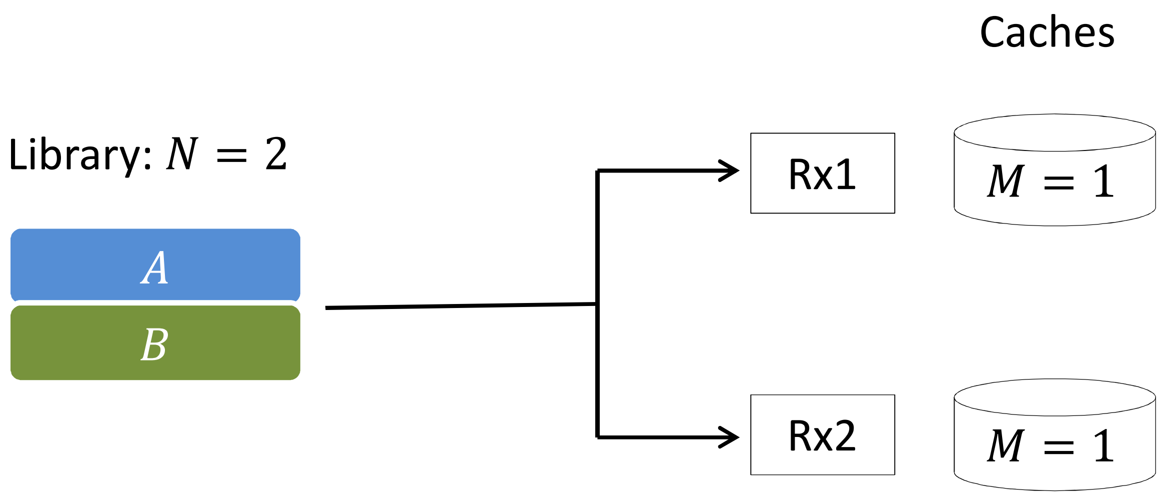


Key breakthrough:

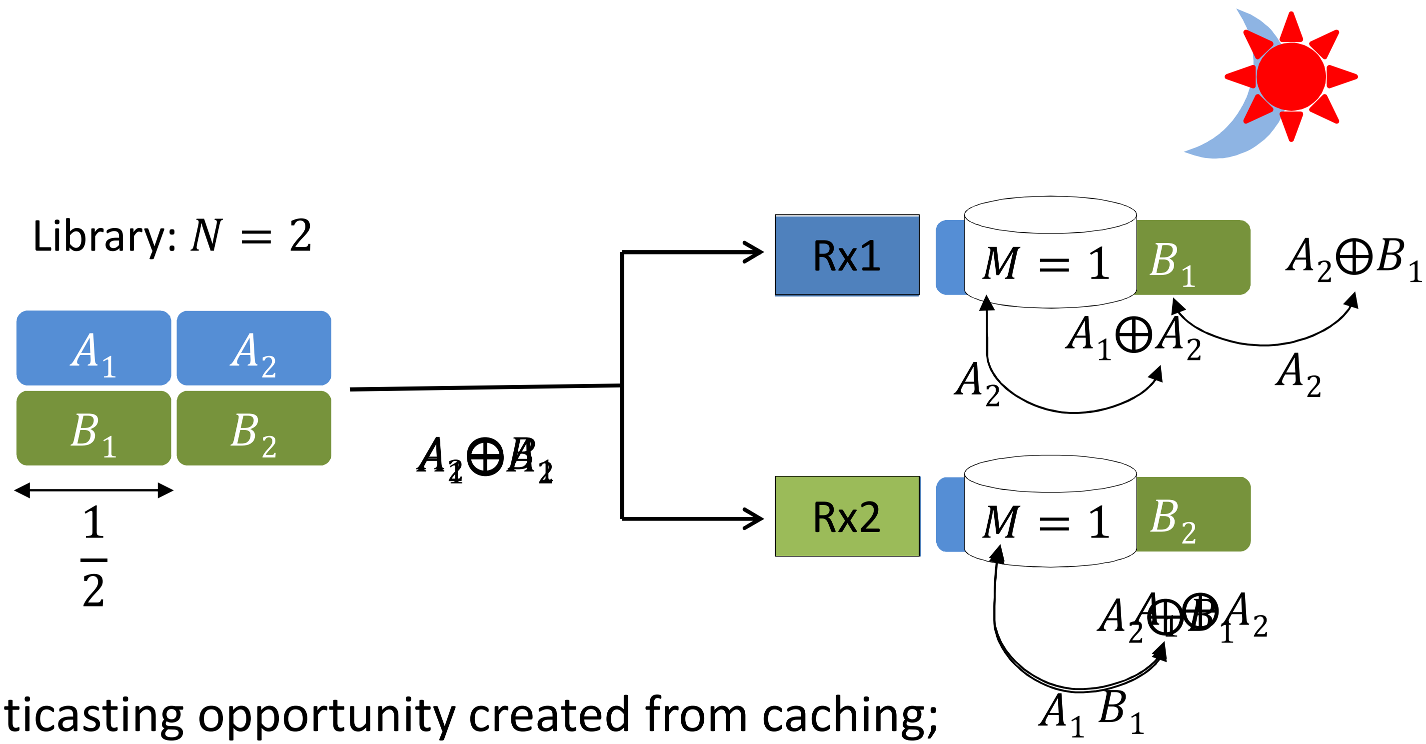
- Cache so that one transmission is useful to many
  - Even if requested files are different
  - Increases multicast opportunities
- Substantial increase in throughput (“worst case”)

Result: Maddah-Ali, Niesen (2012)

Example: $N = K = 2, M = 1 \quad (\gamma = \frac{1}{2})$

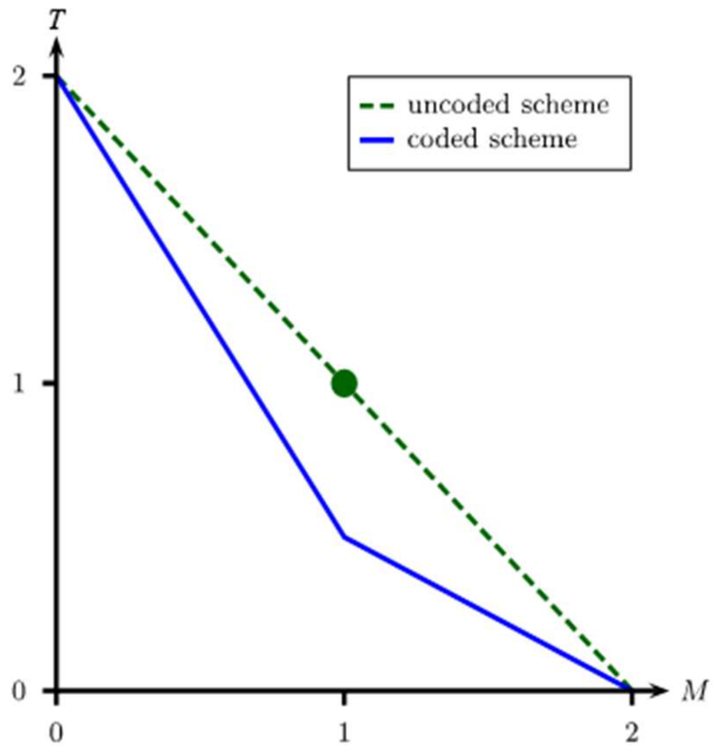


Example: $N = K = 2, M = 1 \qquad (\gamma = \frac{1}{2})$



- Multicasting opportunity created from caching;
  - Hard case: distinct requests
  - Easy case: same requests

Comparison: $N = K = 2, M = 1 \ (\gamma = \frac{M}{N} = \frac{1}{2})$



- Uncoded Caching rate:

$$T: K = 2 \rightarrow K(1-\gamma) = 2 \times \frac{1}{2} = 1$$

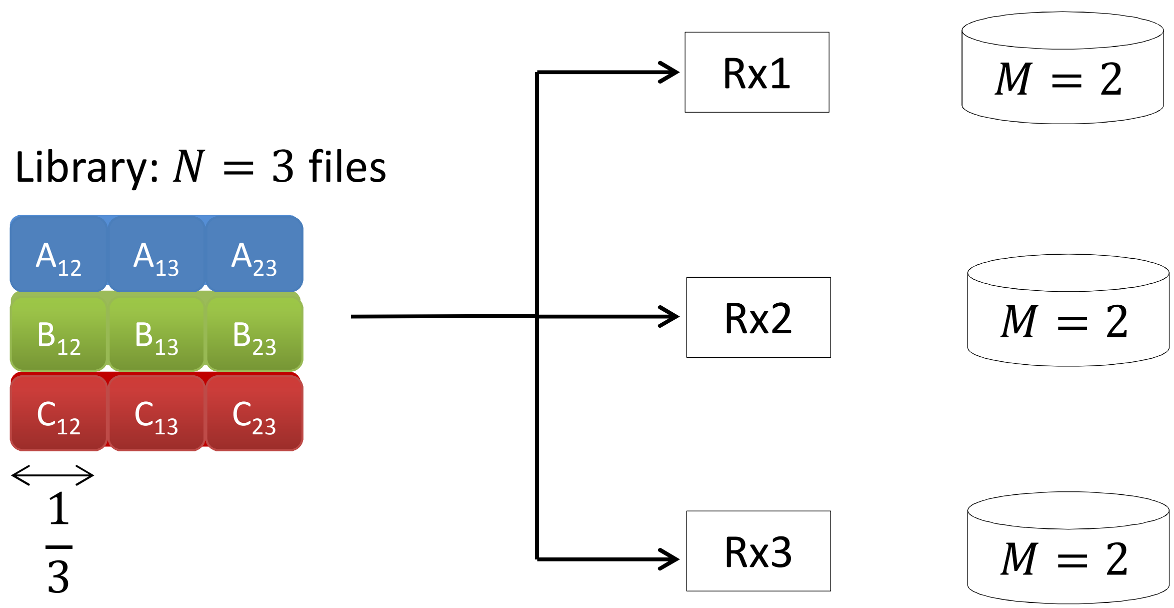
- Coded Caching:

$$T = \frac{1}{2}$$

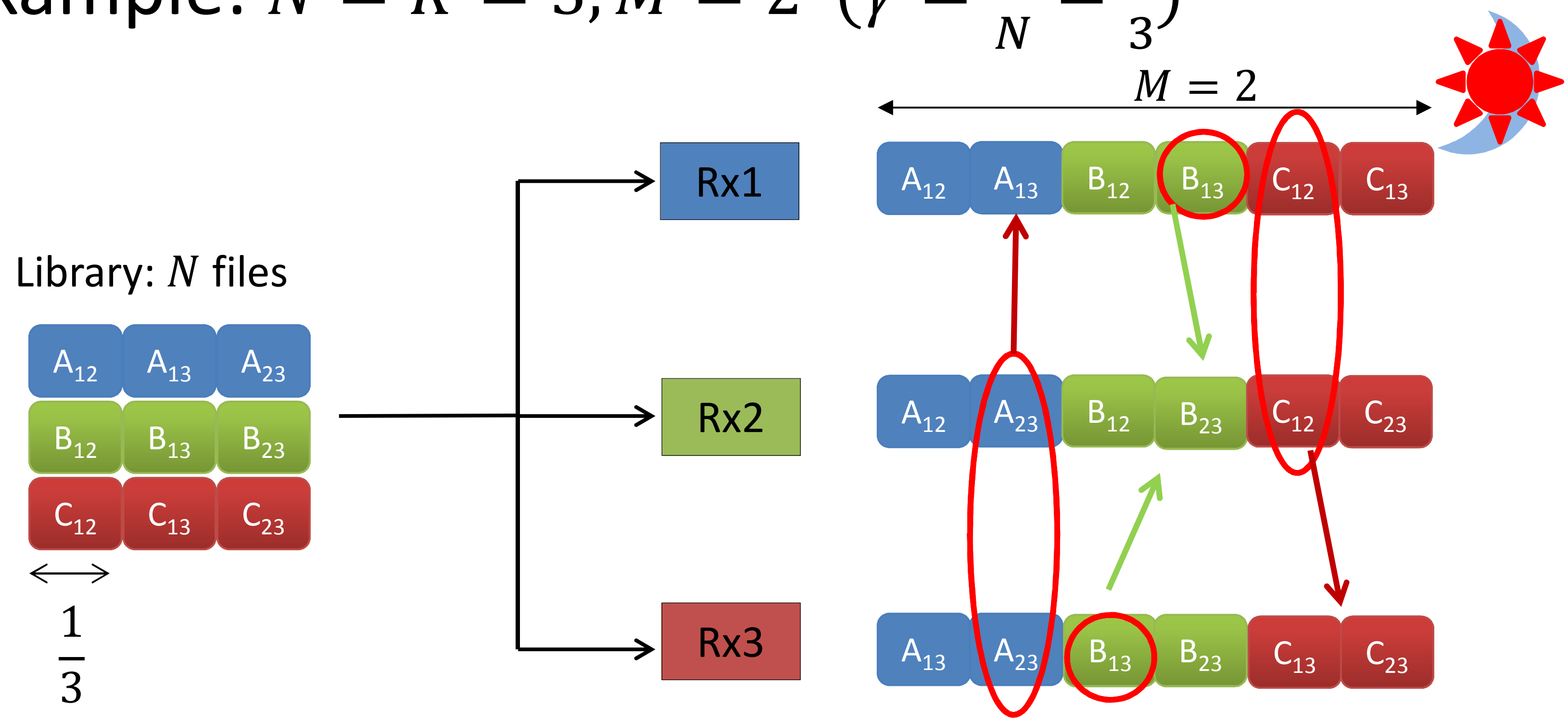
- For $N = K = 2$ case, optimal rate can be achieved for $M \in [0,1]$

Image source: Maddah-Ali, Niesen (2012)

Another Example: $N = K = 3, M = 2$ $\quad (\gamma = \frac{2}{3})$



Example: $N = K = 3, M = 2$ $\left(\gamma = \frac{M}{N} = \frac{2}{3}\right)$

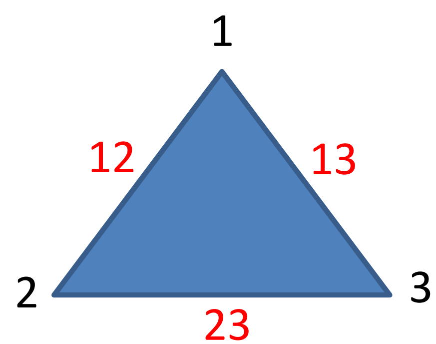


- Transmit : $A_{23} \oplus B_{13} \oplus C_{12}$ (a common message for all)

$$\boxed{T = 1 \times \frac{1}{3} = \frac{1}{3}}$$

Coded Caching Pseudocode (recall $\gamma \stackrel{\text{def}}{=} \frac{M}{N}$)

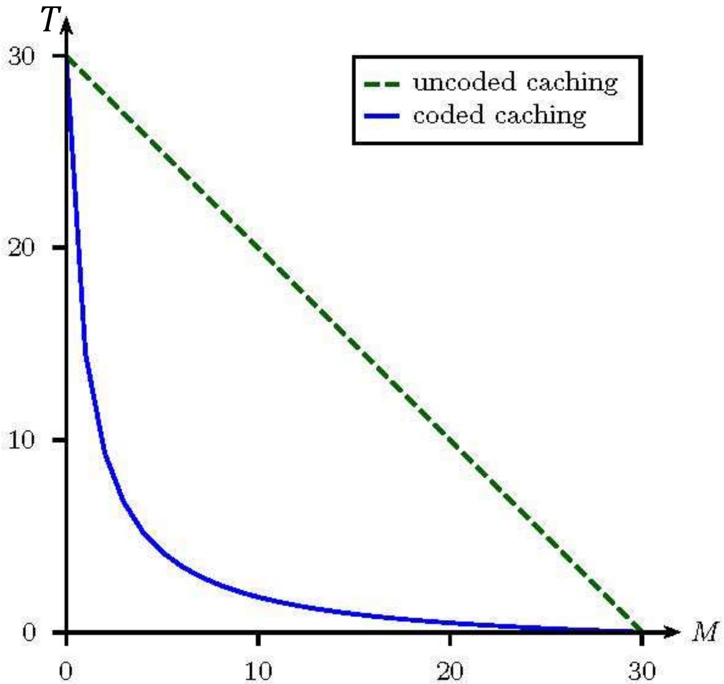
- N files in library
- Split each file into $\binom{K}{KM/N} = \binom{K}{K\gamma}$ subfiles
- *Cache: In every* $\frac{MK}{N} = K\gamma$ *set of users, there is one part of each file in common*



- *Request: Each user asks for one file (out of* N*)*
- *Deliver to* $K\gamma + 1$ *users at a time*
  - *Via XORs with* $K\gamma + 1$ *subfiles.* ***Each user (out of the*** $K\gamma + 1$ ***now served) knows all summands except one (its own requested subfile)***
- *Repeat for all possible sets of* $K\gamma + 1$ *users*

Algorithm: Maddah-Ali, Niesen (2012)

Maddah-Ali and Niesen's results



$N = K = 30$

- Uncoded rate (local caching gain) :

$$T = K(1-\gamma)$$

- Coded-caching required :

$$T = \frac{K(1-\gamma)}{1+K\gamma}$$

  - Coding gain:

$$\text{Gain} \stackrel{\text{def}}{=} \frac{K(1-\gamma)}{T} = 1 + K\gamma$$

Optimal to within a factor 12.

Result and image source: Maddah-Ali, Niesen (2012)

When is coded caching worth the effort?

$K = 10, \gamma = 0.01$ $(K\gamma = 0.1)$:

$T(M) = 9.9$	(only local gain - prefetching)
$T_D(M) = 9.466$	(decentralised caching)
$T_C(M) = 9.0$	(centralised caching)
$T^*(M) \geq 9.0$	(MN optimal bound)

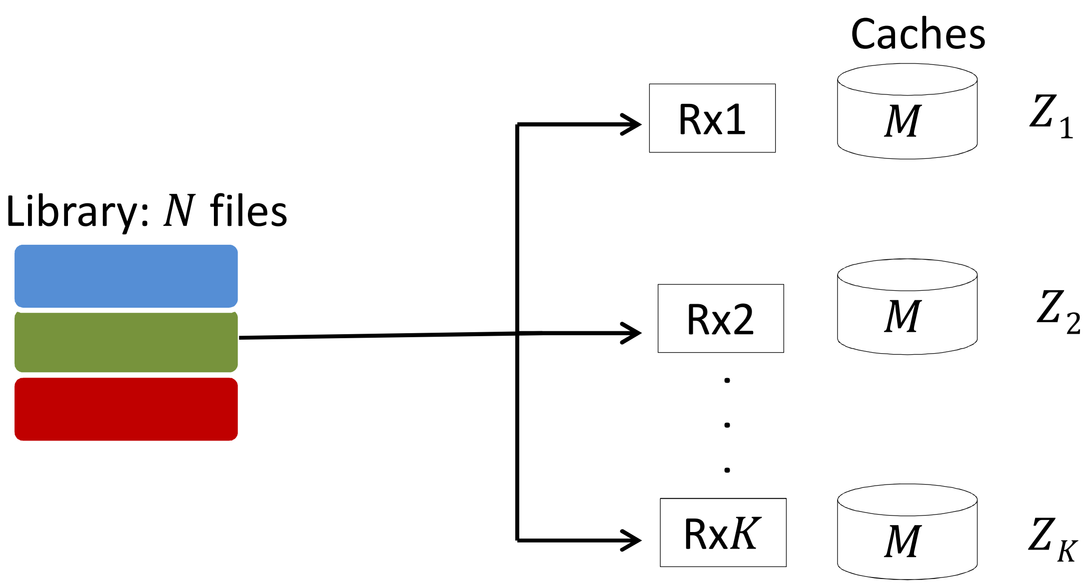
$\Rightarrow$ **Generally small gains when $\boldsymbol{K\gamma < 1}$**

$K = 1000, \gamma = 0.01$ $(K\gamma = 10)$:

$T(M) = 990$	$T_C(M) = 90$
$T_D(M) = 99$	$T^*(M) \geq 25$

Generally large gains when $\boldsymbol{K\gamma > 1}$

On the Optimality of Uncoded Cache-Placement



- Maddah-Ali and Niesen's coded caching is optimal under
  - the constraint of uncoded cache placement

result: Wan et al. (2015)
Maddah Ali et al. (2016)

First Conclusions

- Significant gain of coded caching
  - Treating $K\gamma + 1$ users at a time
  - Worth it when $KM > N$ (unlike traditional caching: $M \approx N$)
- Significant improvement over conventional caching schemes
  - For large K, then T need not scale as K

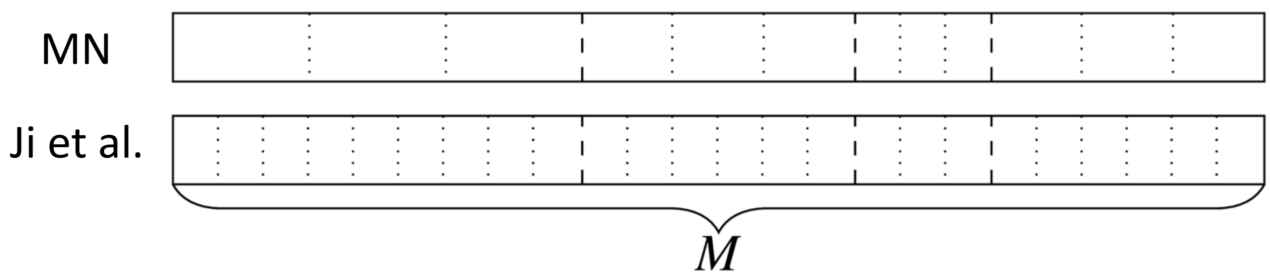
$$T \approx \frac{1-\gamma}{\gamma} \approx \frac{N}{M}$$

- Potential bottlenecks for small γ: T increasing sharply as γ decreases

Coded Caching with Non-uniform Demands

Index-Coding based Scheme for Non-Uniform Demands

- Subfile size same for all files
- Popular files get more subfiles
- Improvement by creating coding opportunities between batches



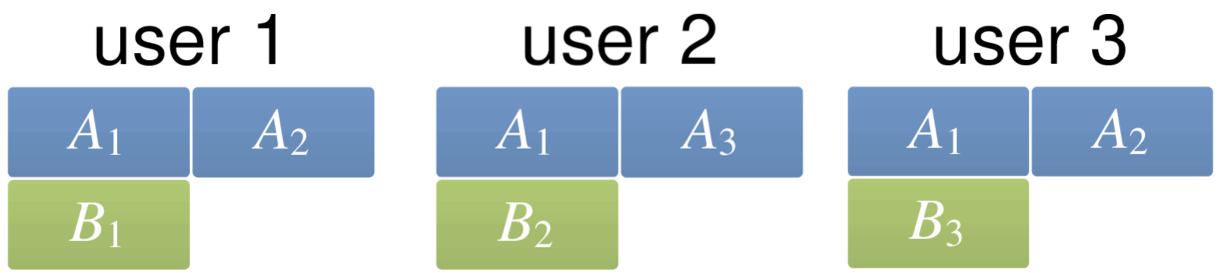
- Delivery uses index coding to combine (XOR) different subfiles
  - graph coloring
  - clique cover

Source: Ji-Tulino-Llorca-Caire (2015)

Example

- 3 files $\{A, B, C\}$ split into 3 parts each. E.g. $A = \{A_1, A_2, A_3\}$
- Cache distribution $\boldsymbol{p} = \{A = \frac{2}{3}, B = \frac{1}{3}, C = 0\}$

Cache realization $\mathcal{C}$



Request: user1→ A, user2→ B, user3→ C

Queried parts: $\mathcal{Q} = \{A_3, B_1, B_3, C_1, C_2, C_3\}$

Conflict Graph $H_{C,Q}$

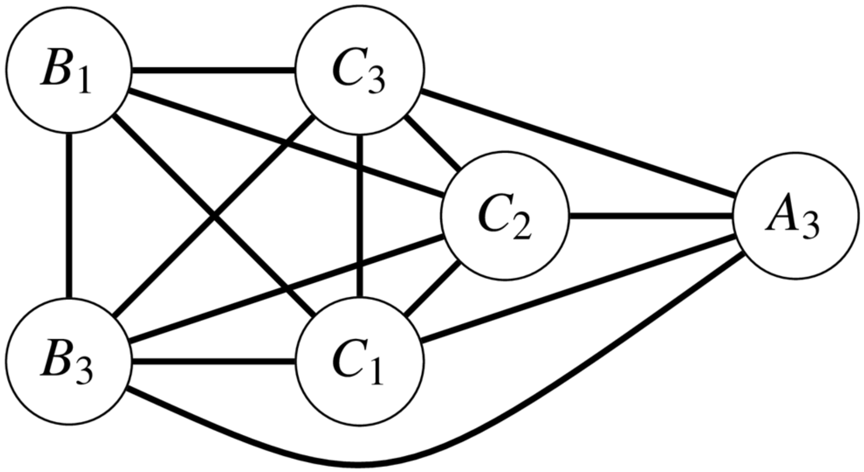
Vertex for each requested subpart ($\in \mathcal{Q}$):

- Replicate if multiple requests of a subfile

Edge if

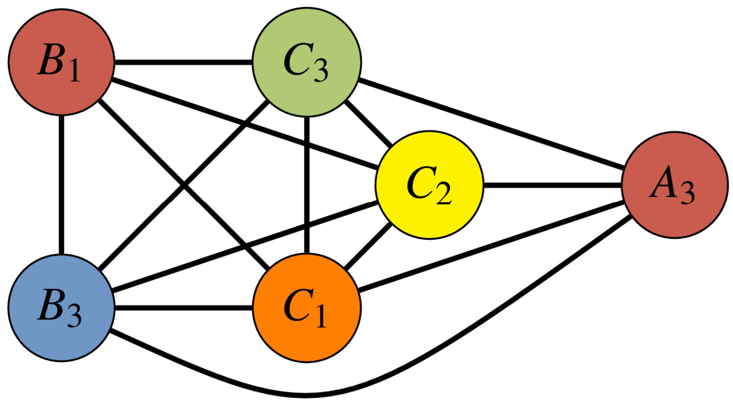
- Not same identity (cannot connect subfile to itself)
- Request(er) not among users caching the other vertex
  - see (A_3, B_1)

Requests: user1→ A, user2→ B, user3→ C
Queried parts: $\mathcal{Q} = \{A_3, B_1, B_3, C_1, C_2, C_3\}$



Graph Coloring $H_{C,Q}$

Connected vertices must have different colors



Transmission

$A_3 \oplus B_1$ $\quad B_3 \quad C_1 \quad C_2 \quad C_3$

$$\boxed{T(\gamma) = 5/3}$$

Gain

$$\frac{|Q|}{\chi(H_{C,Q})}$$ (χ is chromatic number)

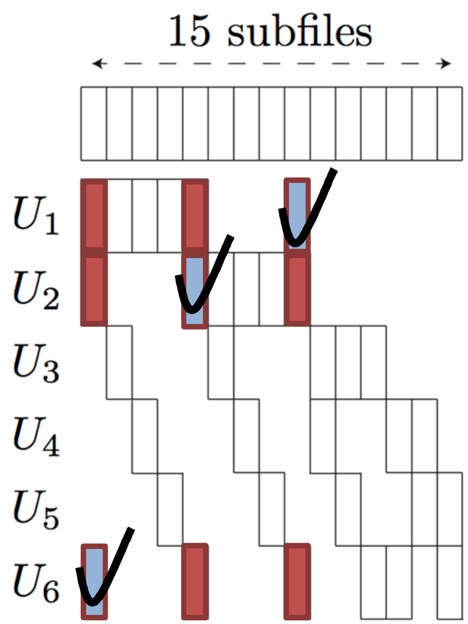
<u>Calculation</u>:

$$\frac{|Q|}{\chi(H_{C,Q})} = \frac{K(1-\gamma)}{T} = \frac{3\left(1-\frac{1}{3}\right)}{5/3} = \frac{6}{5}$$

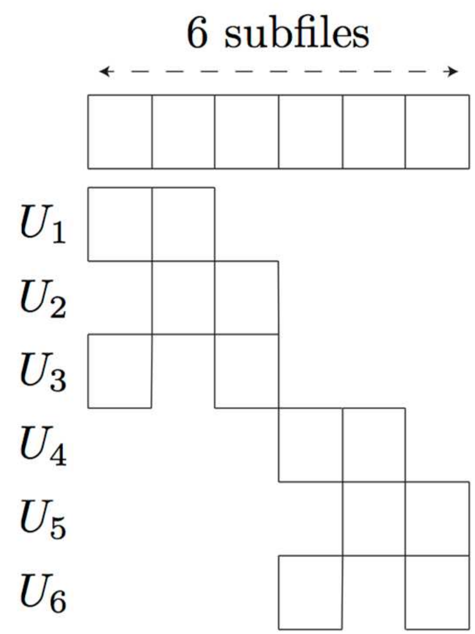
Achilles Heel of Coded Caching

Subpacketization Problem
(Motivates Fusing Coded-Caching and PHY)

$K = 6, K\gamma = 2$



VS



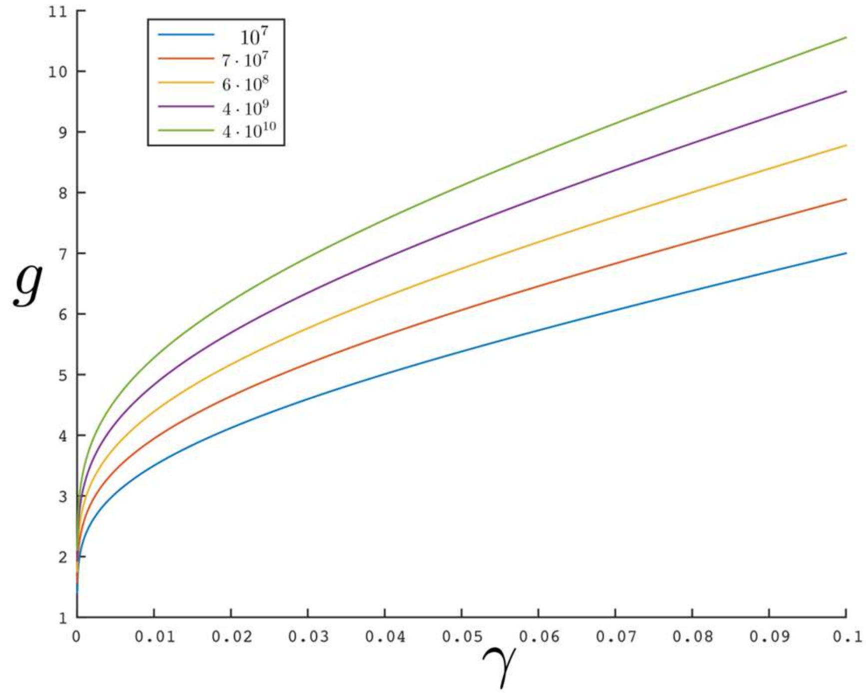
Users are served $K\gamma$+1 at a time

No multicasting between, 1-2-6

Users 1,2,6 get content simultaneously

No overlaps between 1-6 and 2-6

Effective gains under subpacketization constraints



$\binom{K}{K\gamma}$ subfiles

- Maxed over all K
- For original decentralized scheme: gain ≤ 2 if Subpacketization $\leq \frac{e^{K\gamma}}{K\gamma}$

New developments in reducing subpacketization constraints

- Interest in designing algorithms that can tradeoff gain with subpacketization costs

- First breakthrough: Yan et al. (2015) (also Tang et al. 2016)
  - **Placement delivery array** approach
  - Uses Zig-Zag codes from distributed storage (Tamo-Wang-Bruck)

Previous (MN)

$$gain = K\gamma + 1 \qquad Subpacketization \approx \left(\frac{e}{\gamma}\right)^{K\gamma}$$

New (Yan et al.)

$$gain = K\gamma \qquad Subpacketization = \left(\frac{1}{\gamma}\right)^{K\gamma-1}$$

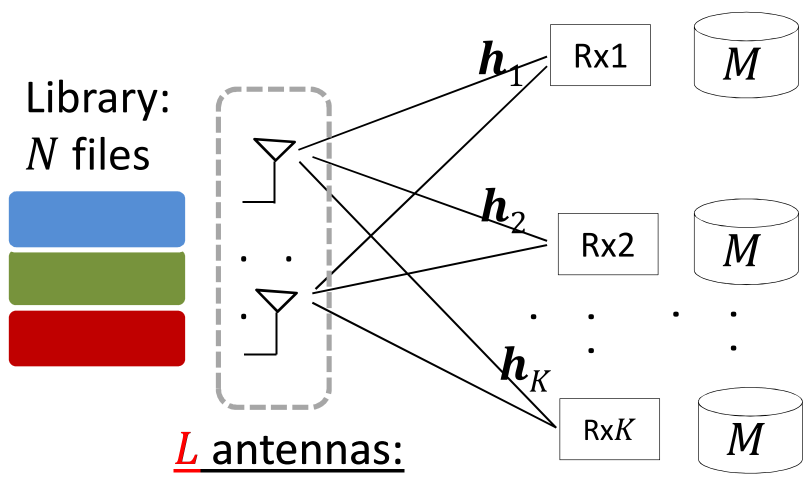
- Some limitations on the available values of γ

New developments in reducing subpacketization constraints

- Shangguan et al (2016). Hyper-graph theoretic approach:
  - *"There do not exist caching schemes that achieve a constant T with subpacketization that grows linearly with K."*
  - Interesting constructions that tradeoff performance with subpacketization
  - Need $K > \frac{4}{\gamma^2}$ (approximately) to get $\text{gain} \geq 2$
    - Reduced coding gain $\approx \frac{K\gamma^2}{4} \ll K\gamma$
  - K must (essentially) be a square integer (thus rarer for $gain \geq 2$):

- Shanmugam et al. 2017 (employed Ruzsa-Szemeredi graphs):
  - *"Gain can scale (suboptimally) with K, with subpacketization that scales almost linearly with K"*
  - *Interesting result of a theoretical nature*
    - Problem: $T < K$ needs massively large $K \gg 1$

Bottlenecks Introduce Need to Combine Memory and PHY Resources in Wireless Networks



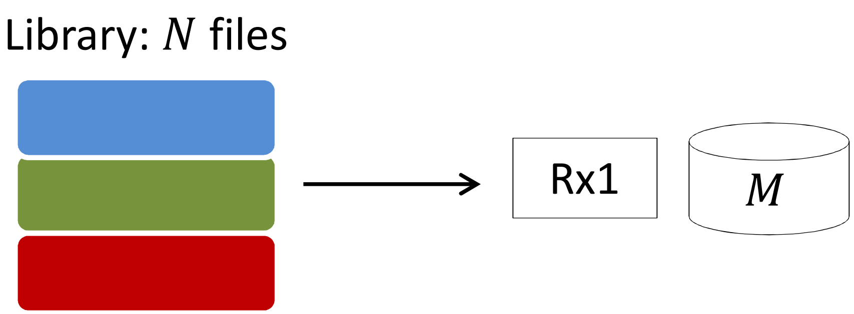
- Exploit additional important resources
  - Linear combinations on the air
  - MIMO
  - Feedback
- Take advantage of salient features of wireless
  - Non linearities
  - Topology
  - Spatial reuse

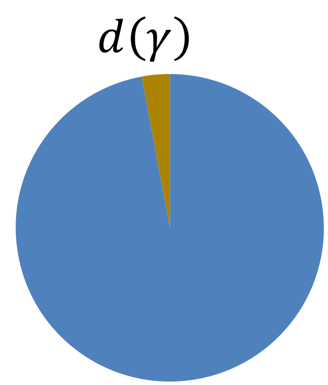
(Cache-aided Degrees of Freedom)

- A equivalent measurement: per-user DoF

$$d(\gamma) = \frac{1-\gamma}{T} \in [0,1]$$

  - $\gamma = \frac{M}{N}$ is normalized local caching gain: prefilled content
  - $Kd(\gamma)$ is the gain

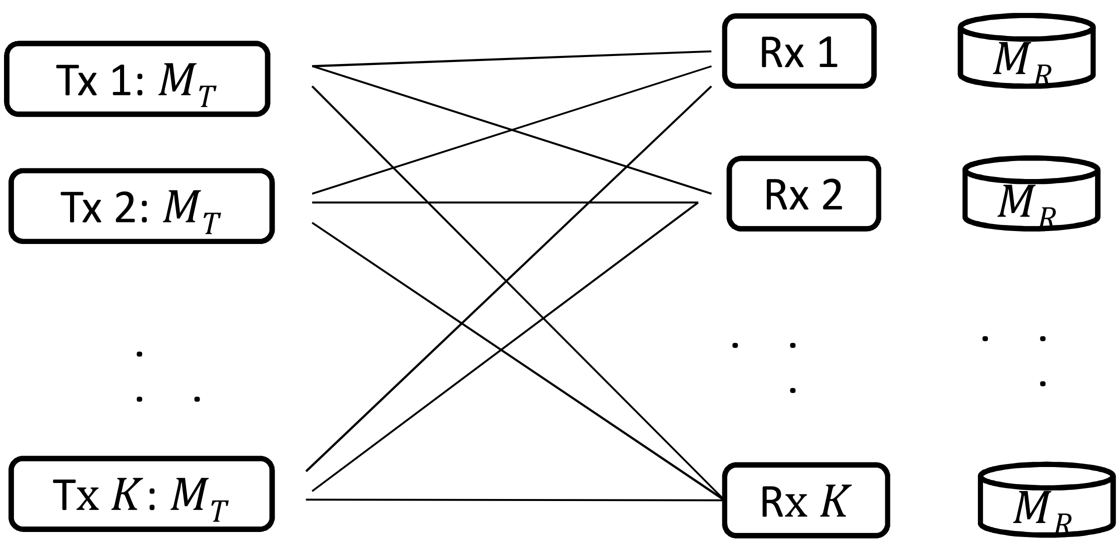




- $T_{opt} = 1 - \gamma \;\Rightarrow d(\gamma) = 1$ (interference-free)

One Shot Cache-aided Interference channel

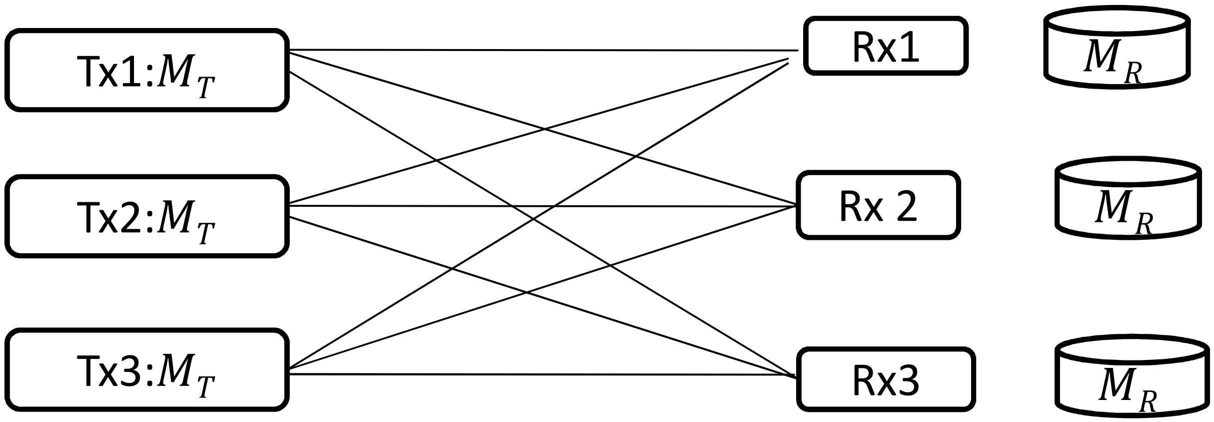
- Cache-aided interference channel
    - K interfering transmitter/ receiver pairs (fully connected)
    - Each transmitter has cache with size $M_T < N$ $(\gamma_T \overset{\text{def}}{=} \frac{M_T}{N})$
    - Each receiver has cache with size $M_R < N$ $(\gamma_R \overset{\text{def}}{=} \frac{M_R}{N})$



Note:
$KM_T \geq N$

Result: Naderializadeh et al. (2016)

Example: $N = K = 3, M_T = 2, M_R = 1$



- N files: $\mathrm{W}_1 = A, \mathrm{W}_2 = B, \mathrm{W}_3 = C$; $\quad (\gamma_T = \frac{M_T}{N} = \frac{2}{3}, \gamma_R = \frac{M_T}{N} = \frac{1}{3})$
- Split each file into $\binom{K}{K\gamma_T}\binom{K}{K\gamma_R} = \binom{3}{2}\binom{3}{1} = 9$ parts

$$A = (A_{12,1}, A_{12,2}, A_{12,3}, A_{13,1}, A_{13,2}, A_{13,3}, A_{23,1}, A_{23,2}, A_{23,3})$$

- Cache Tx 1: $A_{\mathbf{1}2,1}, A_{\mathbf{1}2,2}, A_{\mathbf{1}2,3}, A_{\mathbf{1}3,1}, A_{\mathbf{1}3,2}$
- Cache Rx 1: $A_{12,\mathbf{1}}, A_{13,\mathbf{1}}, A_{23,\mathbf{1}}$

Source: Naderializadeh et al. (2016)

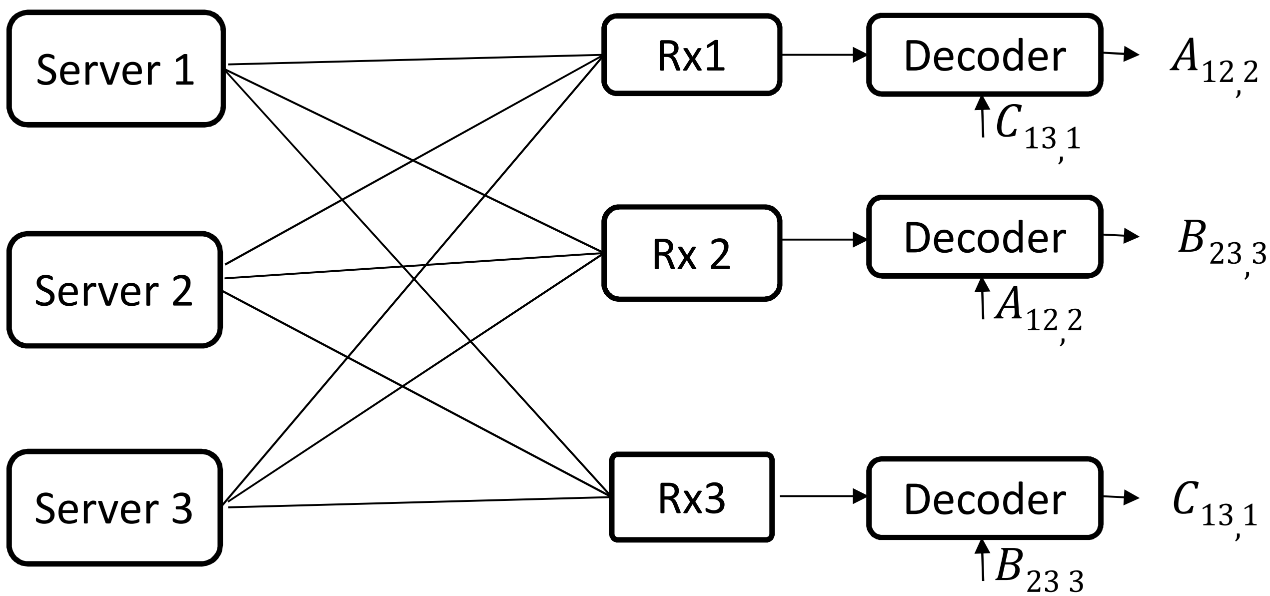
Example: $N = K = 3, M_T = 2, M_R = 1$

- Rx1 needs: $\boxed{A_{12,2}}, A_{12,3}, A_{13,2}, A_{13,3}, A_{23,2}, A_{23,3}$
- Rx2 needs: $\boxed{B_{23,3}}, B_{13,1}, B_{12,3}, B_{23,1}, B_{13,3}, B_{12,1}$
- Rx3 needs: $\boxed{C_{13,1}}, C_{23,2}, C_{23,1}, C_{12,2}, C_{12,1}, C_{13,2}$

$X_1 = L(\boxed{A_{12,2}}, \boxed{C_{13,1}})$

$X_2 = L(\boxed{A_{12,2}}, \boxed{B_{23,3}})$

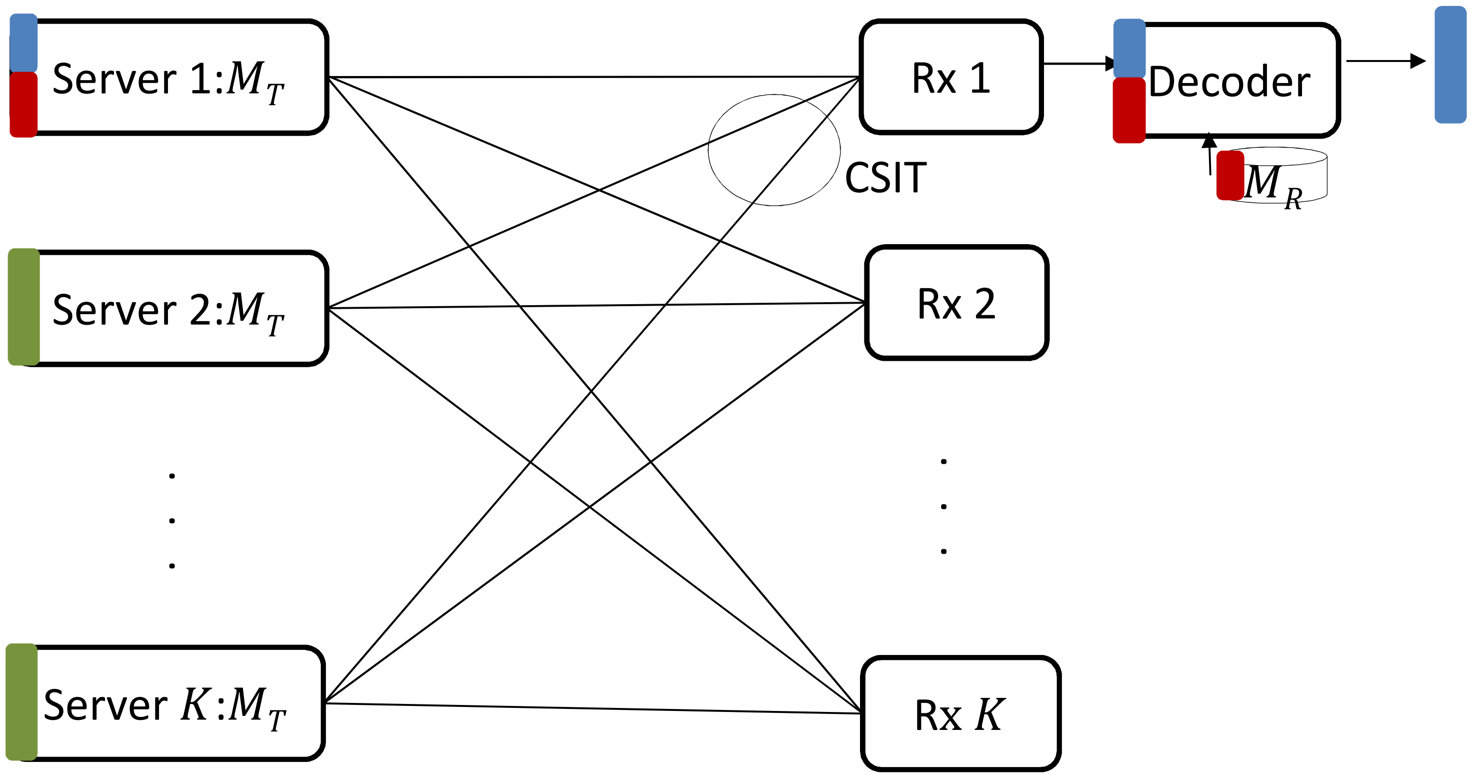
$X_3 = L\underbrace{(\boxed{B_{23,3}}, \boxed{C_{13,1}})}_{1/9}$



- Other triple symbols are the same: $T = \frac{2}{3} \Rightarrow$ $d_\Sigma = \frac{K(M_T+M_R)}{N} = 3$

Source: Naderializadeh et al. (2016)

Idea for the General Case



- With transmitter cooperation and perfect quality CSIT
  - ➢ interference can be cancelled
- Combining with the caching content
  - ➢ recover the missing information in cache

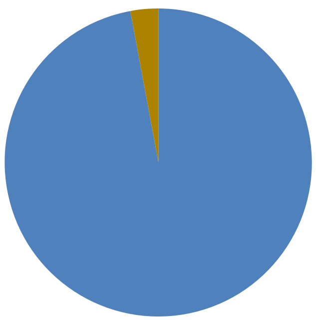
algorithm: Naderializadeh et al. (2016)

Conclusion – Cache Aided IC (one shot)

- The one-shot linear sum-DoF:

$$d_\Sigma = K\gamma_T + K\gamma_R \leq K$$

$$d(\gamma_T, \gamma_R) = \gamma_T + \gamma_R \leq 1$$



- ➢ Gap ≤ 2 from one-shot linear-DoF optimal
- ➢ Equal contribution of tx and rx caches (can change: Shariatpanahi 2017)
- ➢ Covers single-stream (MN-13) and multi-server cases (Shariatpanahi et al. 2015, SEE ALSO Shariatpanahi-Caire-Khalaj 2017).
- ➢ Features exploited: sums on the air (MIMO), CSIT

result: Naderializadeh et al. (2016)

Caching and Feedback

Feature to be exploited: MIMO, CSIT, non-linearity

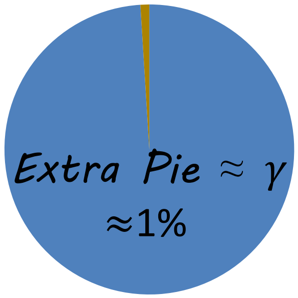
Reveals synergy and interplay

between memory and feedback

Background

- In most cases, DoF impact of coded caching:

$$d(\gamma) - d(\gamma = 0) = \gamma$$



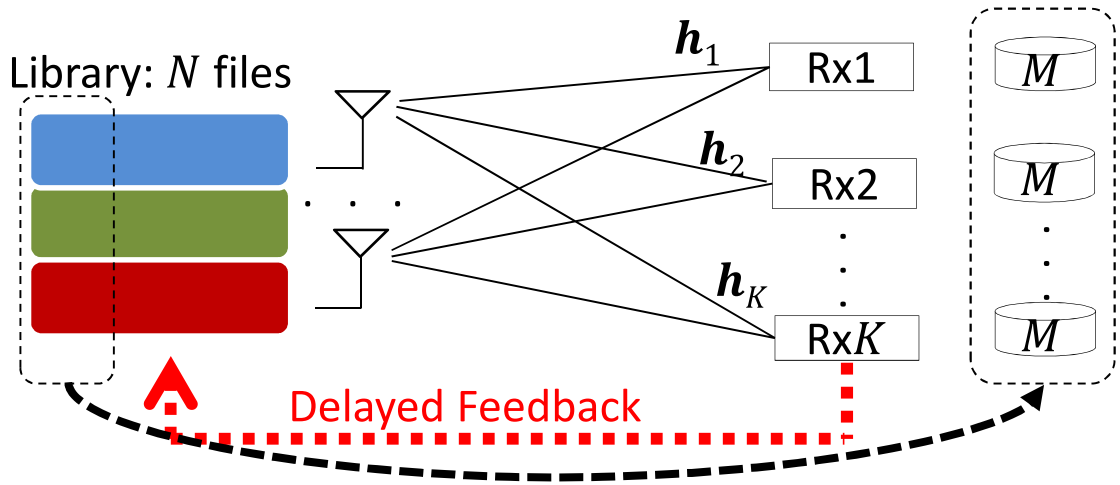
  - Even in settings with perfect feedback and many antennas

<u>Additional</u> "piece of pie" due to caching $\approx \gamma \approx 10^{-3} \rightarrow 10^{-2}$ (Roberts et al.)

- Are there settings for which the impact of caching is substantially larger?

Note – This is a `negative' way of looking at effect of CC

Cache-aided K-user BC with delayed CSIT



- Feature: non-linearity

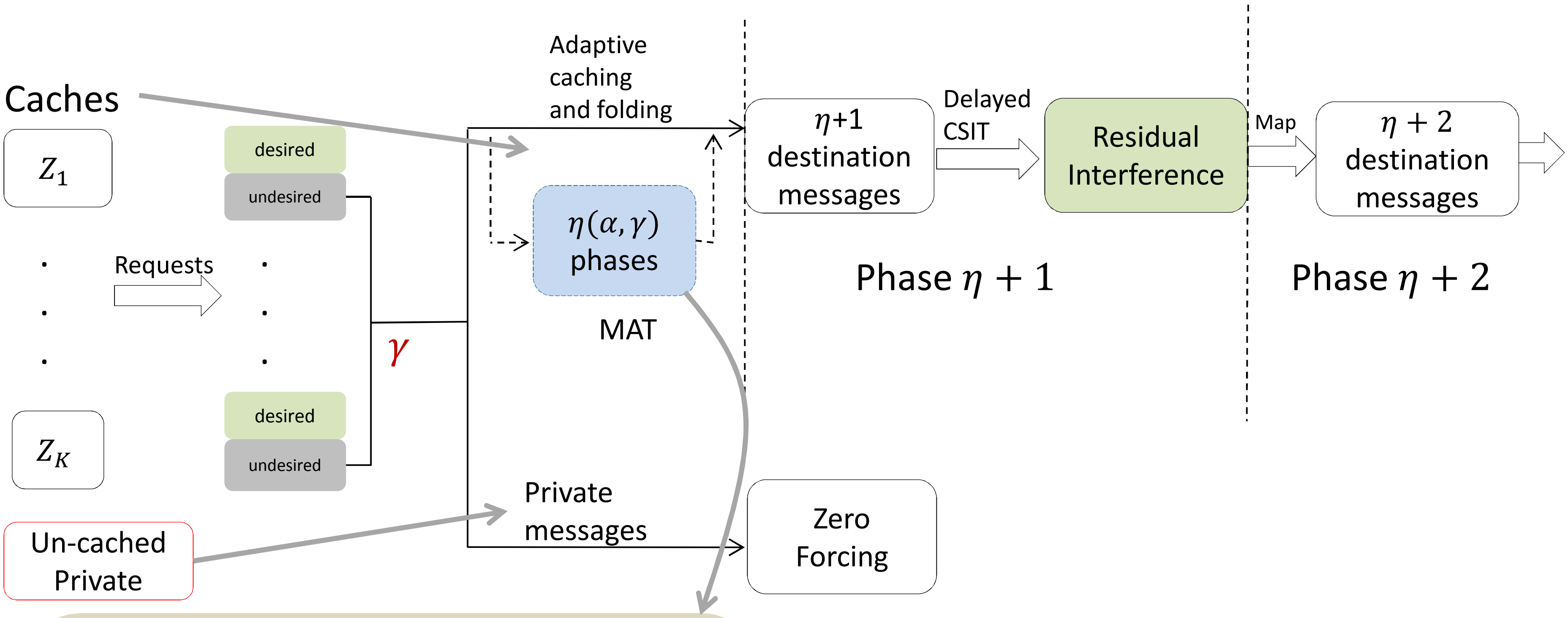
Corollary (Zhang-Elia):

$$T(\gamma) = \log\left(\frac{1}{\gamma}\right) \ll \frac{1}{\gamma}$$

Per-user DoF

$$d(\gamma) = \frac{1-\gamma}{\log\left(\frac{1}{\gamma}\right)}$$

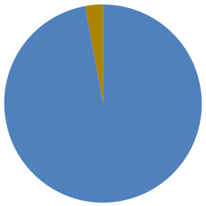
Cache-aided Prospective-hindsight Scheme

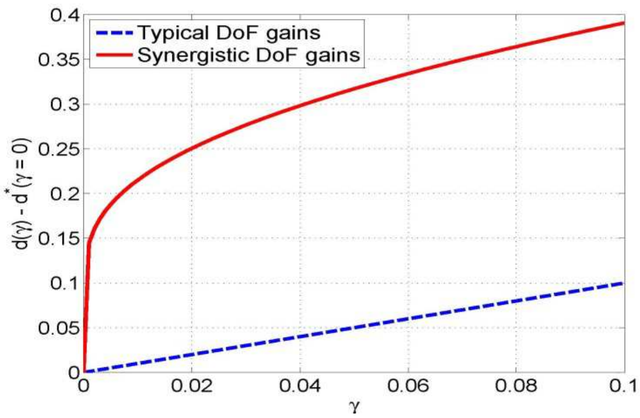


Feature:

- With delayed CSIT, multicasting is much faster than broadcasting
- Memory boosts broadcasting

Synergistic DoF Gains





$$\frac{\delta}{\delta\gamma}\left(d(\gamma) - d(\gamma=0) \right)\Big|_{\gamma=\frac{1}{K}} \approx \frac{K}{\log^2 K} \quad \text{(for all } K\text{)}$$

vs.

$$\frac{\delta}{\delta\gamma}\left(d(\gamma) - d(\gamma=0) \right)\Big|_{\gamma=\frac{1}{K}} = \frac{\delta}{\delta\gamma}(\gamma) = 1$$

- **Feature:** CSIT allows for boost from small (reasonable) amounts of caching

- \`Exponential' effect of coded caching (for sufficiently large K)
  - A very small $\gamma = e^{-G}$ can offer a very satisfactory

$$d(\gamma = e^{-G}) - d(\gamma = 0) \rightarrow \frac{1}{G}$$

Topology (no FB)

Wireless Coded Caching:
A Topological Perspective

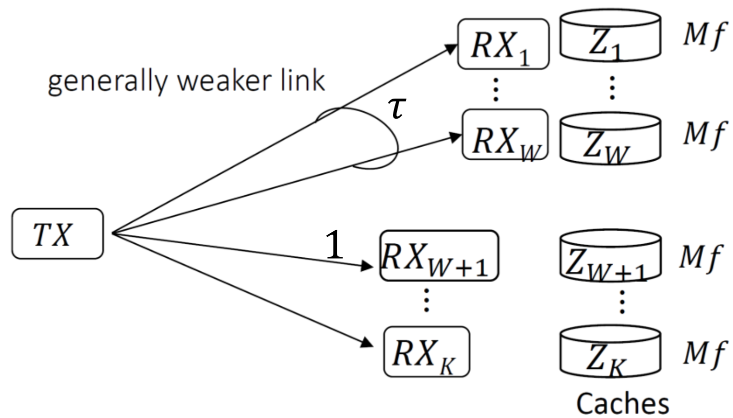
Problem:

- `Worst-user effect: one bad apple…..

Features/Opportunities:

- Topological `holes’ to attenuate interference

Topological SISO BC



Topologically-uneven wireless <u>SISO</u> K-user BC:

- W weak users with normalized capacity $\tau < 1$
- $K - W$ strong users with normalized capacity $= 1$
- Same cache size per user (M)

- Problem: multicasting can suffer from "worst-user" effect

$$d(\gamma) \rightarrow \tau \cdot d(\gamma)$$

Topology Threshold

Corollary (Zhang-Elia 16):

There is a threshold

$$\tau_{thr} \approx 1 - \left(1 - \frac{W}{K}\right)^{g_{max}}$$

which guarantees full-capacity performance

$$T(\tau \geq \tau_{thr}) = T(K)$$

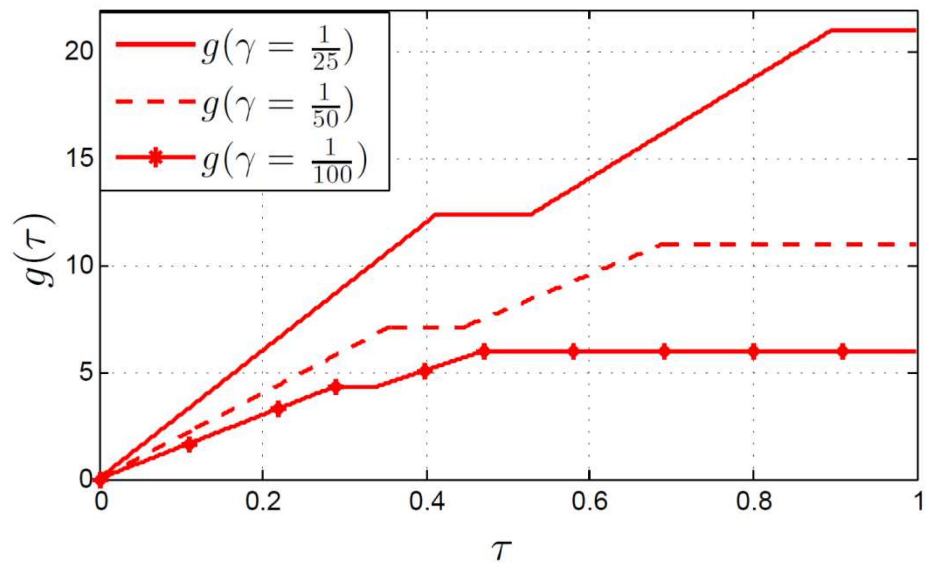
Recall $g_{max} \stackrel{\text{def}}{=} \mathrm{K}\gamma + 1, w \stackrel{\text{def}}{=} \frac{W}{K}$

$$\tau_{thr} \in \left[1 - (1-w)^{g_{max}}, 1 - \left(1 - w - \frac{w\gamma}{1-\gamma}\right)^{g_{max}}\right]$$

Coded-caching Gain

- Coded-caching gain under topology setting

$$g(\tau) \triangleq \frac{K(1-\gamma)}{T} \in [0, g_{max}]$$

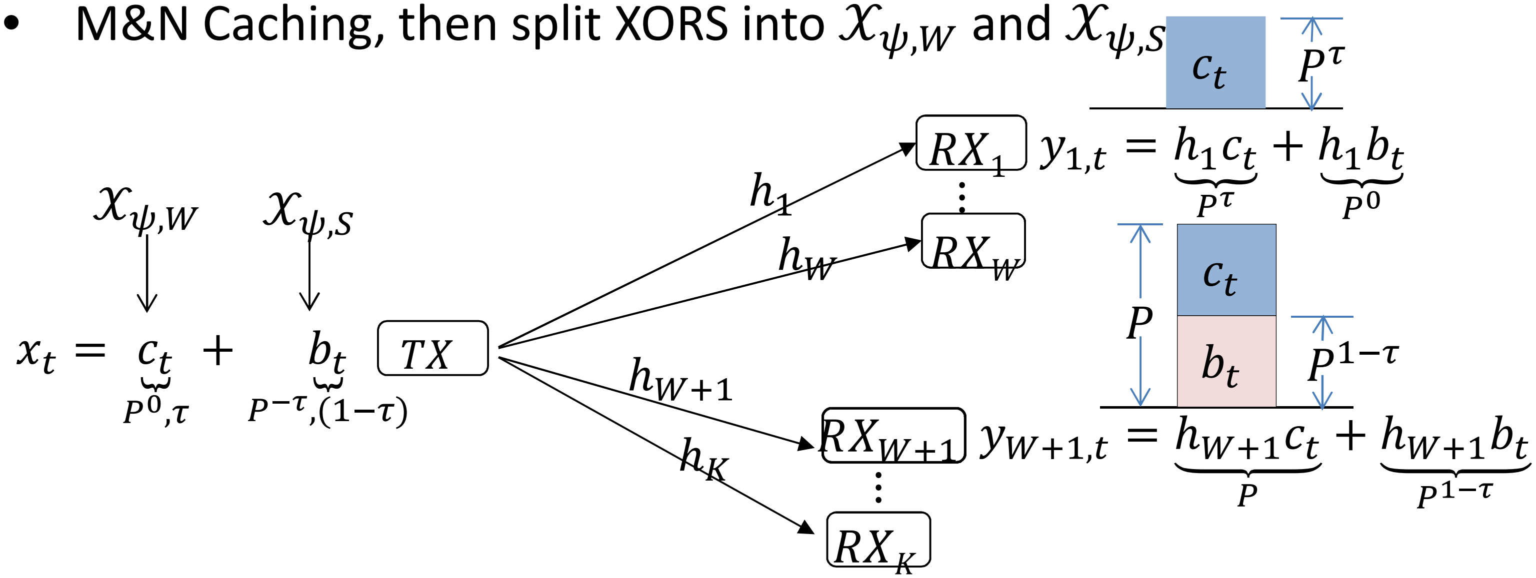


The caching gain for $K = 500, W = 50$

- The horizontal lines denote the maximum gain g_{max} corresponding to $\tau = 1$
- Demonstrate how these can be achieved even with lesser link capacities.

Intuition of the schemes

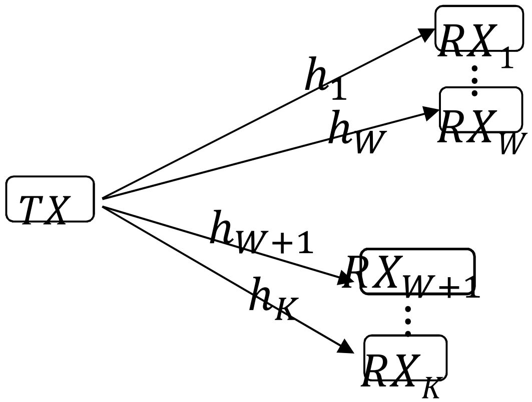
- M&N Caching, then split XORS into $\mathcal{X}_{\psi,W}$ and $\mathcal{X}_{\psi,S}$



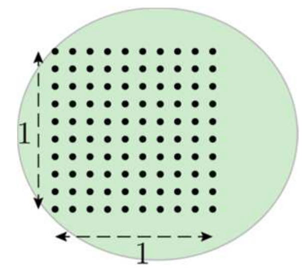
- Interference $\mathcal{X}_{\psi,S}$ hidden from weak users due to topology
  - ➢ Treat strong users ($\mathcal{X}_{\psi,S}$) while slowly serving weak ($\mathcal{X}_{\psi,W}$)
  - ➢ Transmission rate can be kept (in some cases) at 1 (as if all strong)
  - ➢ This ameliorates the negative effects of uneven topology

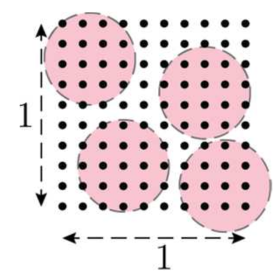
Other salient features of wireless relating to caching

- Topological fluctuations (fading)
  - *"Alpha-fair coded caching"*
  - Salient feature: Channel fluctuation (with power-adaptation, and scheduling) boosts performance and fairness
  - Destounis-Kobayashi-Paschos-Ghorbel 2017



- Spatial Reuse (covering radius of transmitter signals)
  - *"Fundamental limits of caching in wireless D2D"*
  - Salient feature: coded-caching can substitute need for spatial reuse
  - Salient feature: multicasting and spatial reuse are competing resources
  - Ji-Caire-Molisch 2015





General Conclusions

Caching in wireless: recap

- Several salient features when caching is for wireless
  - XORs in the air
  - MIMO
  - Feedback
  - Non linearities
  - Topology
  - Channel fluctuations
  - Spatial reuse…

- Feedback and topology are unexplored frontiers in caching for wireless.
  - Among many interesting differentiating ingredients
  - Key to absorbing structure from data, and transfusing into the channel

- Interesting tradeoffs, synergies, and opportunities
  - Exponential impact of caching
  - Gravidance of Rx vs Tx caches
  - Spatial reuse vs. multicasting
  - Signal separation vs. multicasting
  - Complexity vs. performance

Open Problems and Future Directions

- Fuse Comm-theoretic (info-theoretic) and network theoretic considerations (whatever that means)

- CC in different network topologies
  - Topologies affect FB, interference, and multicasting (all connected)
  - Further ameliorate worst-user effect (progress by Destounis et al.)

- CC in more involved settings
  - E.g. Femto caching ideas with advanced multi-server CC

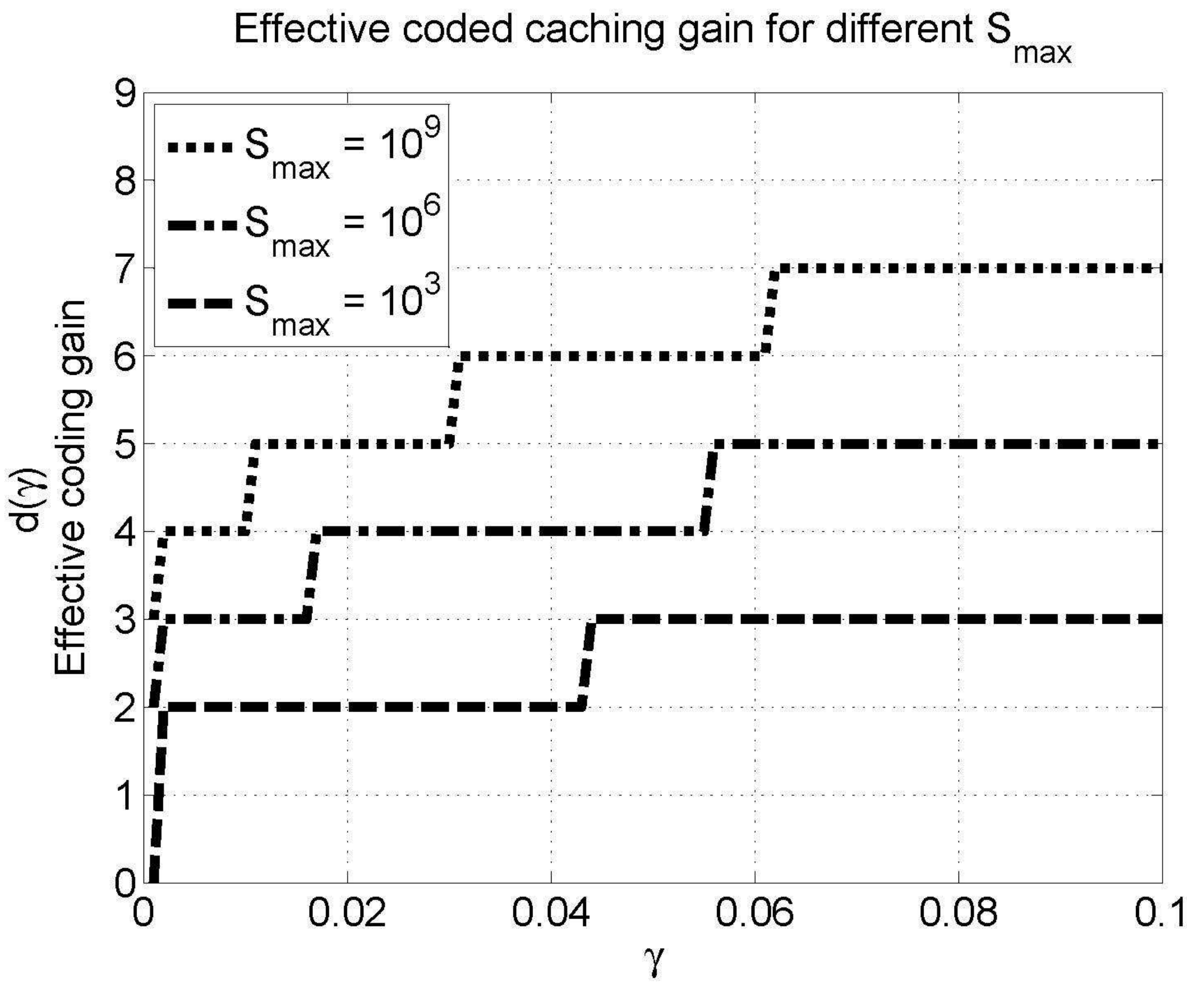
Open Problems and Future Directions

- Caching with secure communications (e.g. https)
  - Public key encryption changes files differently at different receivers
  - (some progress by Paschos et al. and Engelmann-Elia)

- What is the best way to utilize file popularity and user behavior
  - Open problem. Could be key in unlocking CC for commercial use
  - Machine learning: a dual effort to predict channels and requests

- Computational complexity (clique-finding, cache-allocation)

Crippling Bottleneck - Subpacketization



…Stay tuned

Recall theoretical gain $K\gamma + 1 \to \infty$

THANKS FOR YOUR ATTENTION!

**Looking for Postdocs and PhD students